XQC at the Lifelog Search Challenge 2021: Interactive Learning on a Mobile Device

Emil Knudsen
IT University of Copenhagen
Copenhagen, Denmark
emkn@itu.dk

Thomas Holstein Qvortrup
IT University of Copenhagen
Copenhagen, Denmark
thqv@itu.dk

Omar Shahbaz Khan
IT University of Copenhagen
Copenhagen, Denmark
omsh@itu.dk

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bjth@itu.dk

ABSTRACT

In a society dominated by mobile phones and still increasing media collections, Interactive Learning is slowly becoming the favored paradigm for managing these collections. Still, however, no scaling Interactive Learning system exists on a mobile phone. In this paper, we present XQC, an Interactive Learning platform with a user interface that fits most modern smartphones, and scales to large media collections.

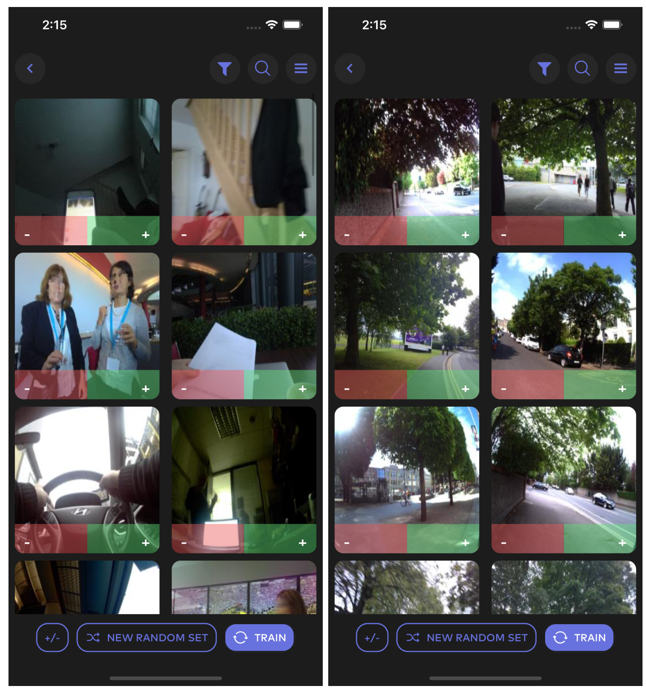
CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; **Search interfaces**; **Retrieval on mobile devices**.

KEYWORDS

Lifelogging; Interactive Learning; XQC.

ACM Reference Format:
Emil Knudsen, Thomas Holstein Qvortrup, Omar Shahbaz Khan, and Björn Þór Jónsson. 2021. XQC at the Lifelog Search Challenge 2021: Interactive Learning on a Mobile Device. In *Proceedings of the 4th Annual Lifelog Search Challenge (LSC '21), August 21, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3463948.3469063



(a) Initial Screen **(b) Classifier Result**

Figure 1: The goal of Interactive Learning is to start from random images (a) and use user feedback to develop a classifier (b). XQC aims to allow Interactive Learning on all platforms with the scalability of the Exquisitor server.

1 INTRODUCTION

The Lifelog Search Challenge [4] (LSC), is an annual competition where different solutions to interactively retrieving multimedia from a multi-modal lifelog are compared [3]. At the LSC competition, researchers are asked to solve search-oriented tasks on a lifelog dataset. Each task consists of a description that is revealed gradually, and a set of relevant images which are considered correct solutions to the task. Even though mobile phones are gaining attraction as the preferred platform for taking images, and the lifelog dataset resembles an enlarged version of the media collection most of us have on our phones, there are no mobile systems participating in LSC.

In recent years, the Interactive Learning paradigm has gained attraction as a valuable addition to the traditional ways of exploring, searching, and filtering large media collections. The goal of Interactive Learning is to make the user and the system collaborate on creating an interactive classifier that best models the user's information needs from a set of random images [5]. This process can be seen in Figure 1 and revolves around labeling suggestions as either positive or negative.

LSC '21, August 21, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8533-6/21/08…$15.00
https://doi.org/10.1145/3463948.3469063

The Exquisitor [6] system is state of the art within Interactive Learning and has competed in previous LSC competitions. However, its current interface does not work on mobile devices. This is why a native Android app, called XQM [1, 8], was developed. XQM handles the questions of scaling down the user interface to mobile clients, but cannot scale to large collections, as mobile phones can only store a relatively limited number of images and the app lacks a high-dimensional index for retrieval performance.

In this paper, we present XQC, a new cross-platform client for Interactive Learning, using the Exquisitor server as a backend. We have taken lessons from XQM regarding scaling down the UI but rely on the scalability of the Exquisitor server to work with large collections. XQC runs on both iOS, Android, and web, and is compatible across all screen sizes.

The outline of the remaining paper is as follows: Section 2 will go through the interactive learning process in the state of the art Exquisitor, with a focus on the backend server. Section 3 will introduce the brand new XQC mobile interface, while Section 4 will explain the solving of LSC tasks using XQC. Lastly, the paper will be concluded in Section 6.

2 EXQUISITOR

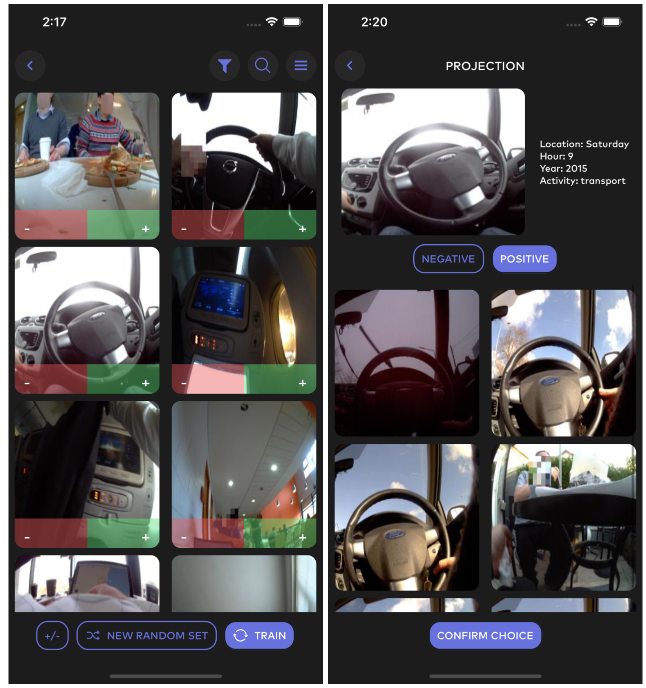
Interactive Learning unfolds in two approaches, an active learning approach, and a user relevance feedback approach. Both approaches aim to make the user and the system collaborate on creating an interactive classifier that best models the user's information needs [5]. While active learning aims to do so by getting the user to label images as close to the decision boundary between classes as possible [7], user relevance feedback systems suggest the items which are ranked the highest based on the previous labeling. In Exquisitor, the system shows the user the highest-scoring media items from the current classifier. The user then gives feedback on all, or some of these items, which is then used to improve the classifier to better represent the user's information need.

Exquisitor utilizes the User Relevance Feedback approach to handle large scale media collections in real time [7]. The system is implemented upon a Linear SVM classifier, that scores media items in a compressed feature space for each interaction round [7]. Moreover, it utilizes a cluster-based high-dimensional indexing strategy called eCP [2].

Exquisitor has a REST API, which allows XQC to scale to LSC standards. After XQC has initialized Exquisitor, the user is shown a random set of images and can start the Interactive Learning process. Once the user is satisfied with her labeling in the current interaction round, she can train the classifier. This action gathers the positively labeled, negatively labeled, and seen images, and sends these images to the Exquisitor, for it to return the improved classifier.

In addition to training the interactive classifier, the Exquisitor REST API allows the user to search for predefined keywords, that are known to be present in the data set. This way, the user can quickly help the model narrow down her information need. This is done by providing a string from the provided dictionary.

The last noteworthy endpoint provided by the API is the filters endpoint. This functionality allows the user to filter out inconsequential media items by providing activities, locations, and time



(a) Projection mode **(b) Image projection**

Figure 2: The use of projection mode. Image (a) shows the main page of interacting with the model. Image (b) shows the 'Projection' functionality that allows the user to see how a labelling affects the classifier.

matching her information need. Exquisitor then saves the preferences and improves the classifier to only contain and present items matching the applied filters.

Lastly, Exquisitor provides a thumbnail server to serve the relevant images in a format that can quickly be rendered.

3 THE XQC MOBILE INTERFACE

XQC was developed using the open-source mobile application framework, React Native. The app is developed with Expo and uses Redux for managing global application state.

When opening the app for the first time, the user gets presented to the welcome screen where users can create a new session or load an existing model. If they choose to create a new session they have to choose what mode they want to build their model with, *Projection* or *Speed* mode. When a session starts for both modes, a set of random images is presented for the user, and from there the user can interact with the model.

3.1 Projection mode (Classifier oriented)

Projection mode is a classifier-oriented mode and was made to cover the need for building an interactive classifier that captures the user's information needs. It provides transparency to the classifier building process, by letting the user see the consequences of each decision they make along the way. This is done by showing the outcome, both if the picture gets labeled positive or negative. Figure 2a shows the initial random images when starting a new session. In this mode, there are two options for labeling images.

The first option is, the user can press on the '-' or ´+' buttons below every image as seen in Figure 2a. This action puts a white border around the button, to indicate whether the image has been labeled as positive or negative. Once the 'train' button has been pressed, the classifier gets trained with the current pool of labeled images.

The second option is, when pressing the image it redirects the user to another screen Figure 2b where they get presented with the outcome of the selected image gets labeled as positive or negative. From here the action can be canceled by going back, or by pressing 'confirm choice' then the classifier is trained with the selected image.

If undoing an action is necessary, the '+/-' button, seen on Figure 2a, can be pressed to deselect already labeled images. The user always gets presented with the current state of the classifier, so to access random images from the Exquisitor, 'new random set' can be pressed.

For both modes, it's possible to open a menu, where the user can save a new model, and overwrite a current saved model. This model is saved in storage, which the user can access at any time until deleted on the load model screen.

3.2 Speed mode (Search-oriented)

Speed mode is a mode focusing on search, and its purpose is finding specific target images as fast as possible. This mode focuses less on transparency and more on speed and competitiveness. In speed mode, the number of images is limited to 6.

When an image is labeled, it immediately trains the model and replaces the labeled image with the current model's image with the best match. This saves time for the user, by not pressing 'train' each time the model has to learn. The 'update' button can be pressed to replace the images with the current state of the model. When starting a new speed session, a timer starts so the user can keep track of their time during a competition.

3.3 Filters and search

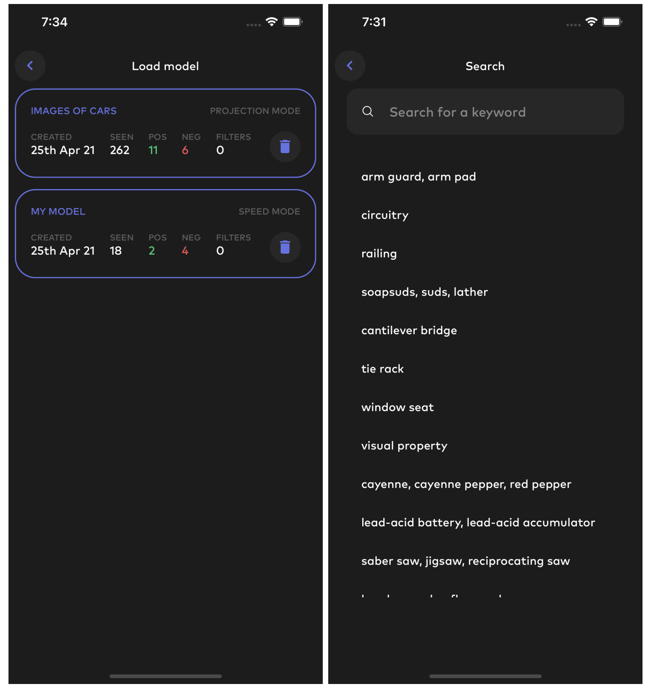
To speed up the process of finding a specific image for a task, we have implemented filter and search functionality to the application. Figure 4b shows how the users can use different parameters to narrow a search down. Here the user can filter on parameters such as activities, locations, year, day, and time. Besides that, a user can also search for specific visual features, which can speed up the process significantly.

3.4 Saving the model

As an extra feature, the app allows users to save a model for later use, as seen in Figure 3a. Saving the model is done by pressing the menu icon in the top right corner of the screen. From this page, the button 'save model as' appears, and from there they can enter a name for the model. The model can then be accessed again from the welcome screen, by pressing 'load model'.

4 SOLVING LSC TASKS

To evaluate the use of XQC, we have solved a couple of tasks from the 2019 competition, they went as follows:



(a) Load model **(b) Search**

Figure 3: Extra features in XQC. Image (a) shows the ability to load previously saved models, that the user can continue working on. Image (b) shows the search functionality, where a pool of 1500 labels can be browsed and chosen to see images with that particular label.

4.1 LSC40

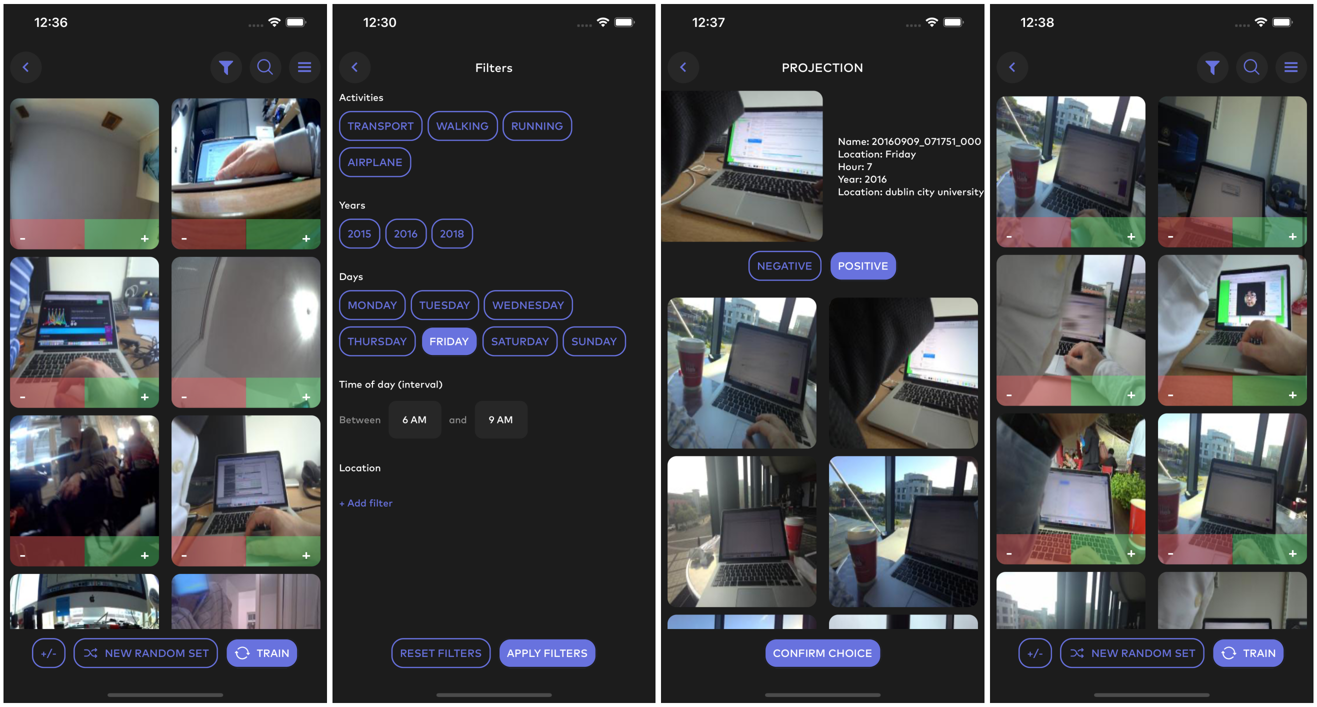
Timestamp 0: There was a racing car. We again started with random images, and had a hard time since no proper results came up when searching for 'racing car'.

Timestamp 30: There was a racing car. It was Formula 1 motor racing. I was at home. This clue helped a lot as we now know that it happens at home and it's on the tv. We used the filters to set the location to home and labeled an image of a tv as positive to train the classifier.

Timestamp 120: There was car racing on TV. It was Formula 1 motor racing. I was at home with two people on a Saturday. The 60 and 90-second timestamps weren't a big help so we used a lot of time finding a relevant image. The next clue stated that the image was taken on a Saturday. This information could be used to filter the images to Saturdays. When we trained the model again with the new filter, we found an image that matched the task query perfectly and we submitted the image to complete the task.

4.2 LSC38

Timestamp 0: I was in my office taking a Skype call. When initializing the model we got a set of random images, and quickly focused on finding an image of a computer, to lead us in the right



(a) Initial Images **(b) Filter** **(c) Projection** **(d) Classifier result**

Figure 4: Steps of solving the task 'LSC38'

direction Figure 4a. After a bit of scrolling, we found an image we deemed interesting and selected it. After reviewing the image's tags and the outcome of labeling the image as both positive and negative we chose to label it as positive since its location was work, which matched the task description.

Timestamp 30: I was in my office taking a Skype call on a Friday morning. When we got the next timestamp, we chose to use the filter where we set the days to Friday and the interval from 6-9 am, to get more precise results Figure 4b.

Timestamp 60: I was in my office taking a Skype call on a Friday morning. I clearly remember a bright screen background on my laptop. A minute into the task, we found an image with green on the background Figure 4c. This meant we were close to finding the relevant image. When the classifier got trained with the image, we found the image called '20160909_085358_000.jpg' and completed the task, as seen on the middle right image on Figure 4d.

5 USER STUDY OF XQC

As part of the development of XQC, a user study was conducted with a focus on the system's ability to solve LSC novice-level tasks. In this study, 8 test persons with no prior knowledge in the Interactive Learning domain were used. They were each given a short introduction to the Lifelog Search Challenge, Interactive Learning, and the newly developed system. They were then asked to solve 2 tasks; one on a mobile phone with one of the modes introduced in Section 3, and one on a computer with the remaining mode.

The study showed very promising results in regards to solving the tasks. The users' first attempt at solving a task was on average 7:55 minutes, with 4/8 test persons conforming to the LSC time limit of 7 minutes for novices. For the users' second attempt a significant improvement was seen with an average of 4:19 minutes and 7/8 test persons successfully submitting a correct image within the 7 minutes.

Furthermore, the study showed only a few seconds of differentiation between platforms. This finding is desirable for XQC as a cross-platform system and verifies that the system is usable and scales as intended to most modern devices as seen in Figure 5.

6 CONCLUSION

In this paper, we have presented the scalable Interactive Learning system XQC. We have gone through the Exquisitor backend and the interface of the newly developed XQC system. By solving previously used LSC tasks, we have proved that Interactive Learning is possible on mobile phones on media collections with the size of the LSC dataset.

ACKNOWLEDGMENTS

This work was supported in part by a PhD grant from the IT University of Copenhagen.

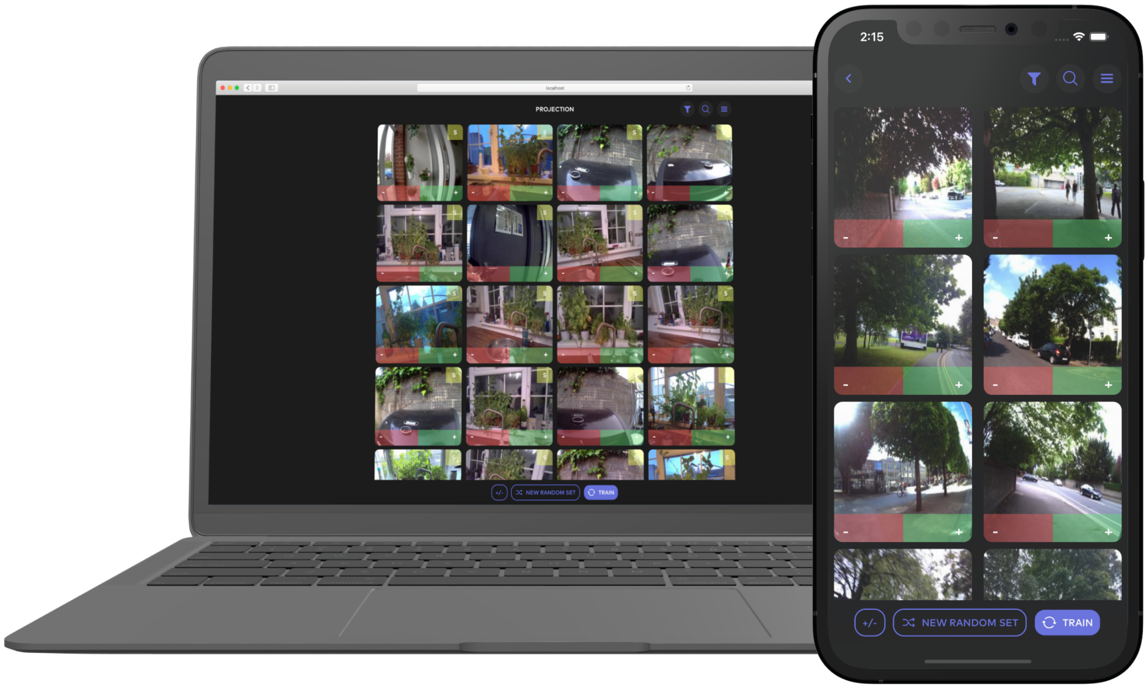


Figure 5: XQC on web and iOS

REFERENCES

[1] Alexandra M. Bagi, Kim I. Schild, Omar Shahbaz Khan, Jan Zahálka, and Björn Þór Jónsson. 2021. XQM: Interactive Learning on Mobile Phones. In *MultiMedia Modeling*, Jakub Lokoč, Tomáš Skopal, Klaus Schoeffmann, Vasileios Mezaris, Xirong Li, Stefanos Vrochidis, and Ioannis Patras (Eds.). Springer International Publishing, Cham, 281–293.

[2] Gylfi Þór Guðmundsson, Laurent Amsaleg, and Björn Þór Jónsson. 2012. Impact of Storage Technology on the Efficiency of Cluster-based High-dimensional Index Creation. In *Proc. International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, Busan, South Korea, 53–64.

[3] Cathal Gurrin, Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn Þór Jónsson, Jakub Lokoč, Wolfgang Hurst, Minh-Triet Tran, and Klaus Schoeffmann. 2020. An Introduction to the Third Annual Lifelog Search Challenge, LSC'20. In *ICMR '20, The 2020 International Conference on Multimedia Retrieval*. ACM, Dublin, Ireland.

[4] Cathal Gurrin, Björn Þór Jónsson, Klaus Schöffmann, Duc-Tien Dang-Nguyen, Jakub Lokoč, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Graham Healy. 2021. Introduction to the Fourth Annual Lifelog Search Challenge, LSC'21. In *Proc. International Conference on Multimedia Retrieval (ICMR'21)*. ACM, Taipei, Taiwan.

[5] Thomas S. Huang, Charlie K. Dagli, Shyamsundar Rajaram, Edward Y. Chang, Michael I. Mandel, Graham E. Poliner, and Daniel P. W. Ellis. 2008. Active Learning for Interactive Multimedia Retrieval. *Proc. IEEE* 96, 4 (2008), 648–667.

[6] Omar Shahbaz Khan, Björn Þór Jónsson, Stevan Rudinac, Jan Zahálka, Hanna Ragnarsdóttir, Þórhildur Þorleiksdóttir, Gylfi Þór Guðmundsson, Laurent Amsaleg, and Marcel Worring. 2020. Interactive Learning for Multimedia at Large. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. Springer, Lisboa, Portugal, 16.

[7] Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2019. Exquisitor at the Lifelog Search Challenge 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC 2019*. ACM, Ottawa, ON, Canada, 7–11.

[8] Magnus Holm Mamsen. 2021. *Interactive Learning On Android Mobile Device.* Master's thesis. IT University of Copenhagen.